# ITNLP-ARC at SemEval-2018 Task 12: Argument Reasoning Comprehension with Attention

**Wenjie Liu, Chengjie Sun, Lei Lin, Bingquan Liu**
School of Computer Science and Technology
Harbin Institute of Technology
Harbin, China
{wjliu, cjsun, linl, liubq}@insun.hit.edu.cn

## Abstract

Reasoning is a very important topic and has many important applications in the field of natural language processing. Semantic Evaluation (SemEval) 2018 Task 12 "The Argument Reasoning Comprehension" committed to research natural language reasoning. In this task, we proposed a novel argument reasoning comprehension system, ITNLP-ARC, which use Neural Networks technology to solve this problem. In our system, the LSTM model is involved to encode both the premise sentences and the warrant sentences. The attention model is used to merge the two premise sentence vectors. Through comparing the similarity between the attention vector and each of the two warrant vectors, we choose the one with higher similarity as our system's final answer.

## 1 Introduction

Reasoning is a very challenging, but basic part of Natural Language Inference (NLI) (Chen et al., 2017), and many relevant tasks have been proposed such as Recognizing Textual Entailment (RTE) and so on. Stanford University provided Stanford Natural Language Inference (SNLI) corpus to support Natural Language Inference task. It contained two kinds of sentences-the premise sentence and the warrant sentence.The mission is to judge whether the two sentences are inference or not. Semantic Evaluation (SemEval) 2018 Task 12-The Argument Reasoning Comprehension-give an argument consisting of the claim, the reason and two warrants. The goal is to select the correct warrant that explains reasoning with this particular argument. There are two options given and only one is correct. Compare with Stanford Natural Language Inference (SNLI) task (Bowman et al., 2015; Shen et al., 2018), it has more challenges. Because it has abundant premise information such as the reason, the claim, text information, as well as the option warrants have high semantic textual similarity (Habernal et al., 2017). In this task, we need to find an effective method to extract important information from these premise sentences.

Natural Language Reasoning can be applied to various fields such as question and answering, information retrieval and so on. With the development of Neural Networks applied in Natural Language Processing, sentence representation and reasoning have been researched and taken significant step forwards. In order to deal with the sequence problem, recurrent neural networks (RNN) (Mikolov et al., 2010, 2011) proposes the concept of hidden state, which can extract features from sequence-shaped data and then convert it to output. It can be used to encode the sentence to fixed-length vector representations. In most recent years, long short-term memory (LSTM) network (Bengio et al., 1994; Hochreiter and Schmidhuber, 1997), BiLSTM (Pennington et al., 2014a) and gated recurrent unit (GRU) (Cho et al., 2014) are widely used to get sentence representative vector, and achieved better result compared with traditional methods. Attention model also known as alignment model pays more attention to two sentences interaction (Zheng et al., 2018; Gao et al., 2018), which is usually applied in information extraction, relation extraction, text summarization and machine translation. In machine translation, the attention model can be focused on one or a few words of input to make the translation more accurate when generating each new word. (Rocktäschel et al., 2015) extend a neural word-by-word attention mechanism to encourage reasoning over entailment of pairs of words and phrases.

In our system, we use long short-term memory network to encode sentence. To make full use of



*Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, pages 1089–1093
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

the information of the reason and the claim, we use attention model to get the attention sentence vector. Then, we compare the warrant sentence vector and the attention sentence vector similarity. The warrant with higher similarity is taken as an answer. In order to make the system more accurate, we use ensemble result as our final answer.

## 2 Method

The dataset composes with four items which are the reason, the claim, the warrant and the alternative warrant (R, C, W, AW), and two additional information: debateTitle and debateInfo. Let R be a reason for a claim C, both of which are propositions extracted from debateTitle and debateInfo. There are two warrants (AW, W) that justify the use of the reason R as support for the claim C. In this task, we choose the correct warrant by these premise information. In our system, we encode sentence with LSTM, and merge two sentences with attention. Then choose the one (AW or W) with higher similarity between the warrant vector and the attention vector as our answer. The system's neural networks model shown as Fig 1. We build the system with five parts, the following is a detailed description.

### 2.1 LSTM

Long short-term memory (LSTM) network is a variant of RNN, and it has been successfully applied to various kinds of NLP tasks. It can solve RNN's problem of gradient vanishing and gradient explosion and be good at dealing with sequence-shaped data. LSTM model controls the memory unit through the input gate, output gate and forget gate. The input is a sequence of sentence $X = \{x_1, x_2, \ldots, x_n\}$, where $x_i$ is the word vector of i'th word in the sentence. The output is $H = \{h_1, h_2, \ldots, h_n\}$, where $h_i$ is the i'th step of the LSTM's output. Here, we use the pre-trained vector of global vectors (GloVe) (Pennington et al., 2014b) as the embedding layer initialization, and the word embedding dimension is 300. The formulas for LSTM include:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (1)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

$$\widetilde{C_t} = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \quad (4)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * tanh(C_t) \quad (6)$$

In our experiment,we encode the reason sentence and the claim sentence with one LSTM encoder, and encode the warrant sentence with another. We try to use the LSTM's last output, mean pooling and max pooling as the sentence vector representation.

### 2.2 Attention

In argument reasoning comprehension task, the claim sentence is extracted from title and information, and it supports the result. Therefore, the claim has a great impact on the reason sentence. So, we use attention model to force the reason's and the claim's similarity word, and get the better premise sentence representation. In this task, we use two kinds of attention model to merge reasons and claims vector representation. Let's $R \in \mathbb{R}^{k \times l_r}$ be a matrix consisting of the reason's LSTM output vector $R = \{r_1, r_2, \ldots, r_{l_r}\}$, and $C \in \mathbb{R}^{k \times l_c}$ be a matrix consisting of the claim's LSTM layer output vector $C = \{c_1, c_2, \ldots, c_{l_c}\}$, where $l_r$ is the length of the reason, $l_c$ is the length of the claim, and k is the LSTM's outputs dimension.

One of the attention model is seq-attention model. In our system, we try to represent the claim sentence vector as $c \in \mathbb{R}^k$, where c is the LSTM's last output, mean pooling or max pooling. Then, calculating the claim sentence vector c and the reason sentence vecotor's $\{r_1, r_2, \ldots, r_{l_r}\}$ similarity as the attention weight. We use the result of two vectors multiplication as the similarity weight. Finally, we can obtain the reason sentence vector with weight. The calculation process is as following:

$$e_i = c \bullet r_i \quad (7)$$

$$\alpha_i = \frac{exp(e_i)}{\sum_{i=1}^{l} exp(e_i)} \quad (8)$$

$$Rtt^* = \sum_{i=1}^{l} \alpha_i \bullet r_i \quad (9)$$

where $\alpha$ is the attention weight. The attention vector represent as $Rtt^* \in \mathbb{R}^k$.

Another kind of attention uses matrix to calculate the weight of the claim sentence vector and the reason sentence vector. Give each sentence vector a weight matrix, and obtain the attention vector by learning the weight matrix. The formula is:

$$M = \tanh(W_y R + W_h C_{l_c} \otimes e_{l_r}) \quad (10)$$

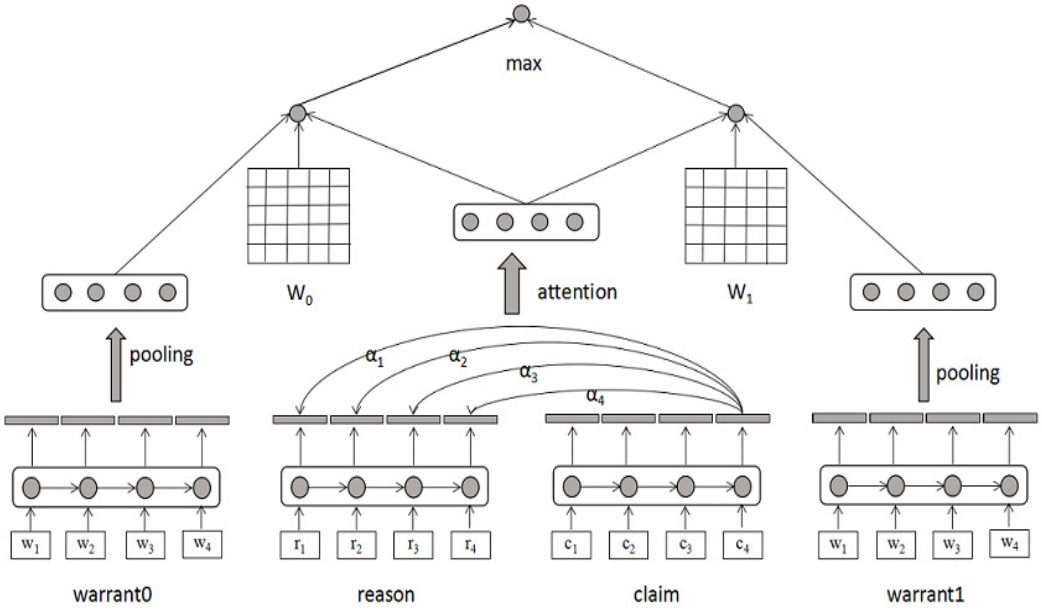


Figure 1: Our system model with attention-encoding with LSTM, merge the reason and the claim with attention, and calculating text similarity with belinear.

$$\alpha = \text{softmax}(w^T M) \tag{11}$$

$$r = R\alpha^T \tag{12}$$

$$Rtt^* = \tanh(W_p r + W_x C_{l_c}) \tag{13}$$

where $W_y \in \mathbb{R}^{k\times k}$, $W_h \in \mathbb{R}^{k\times k}$, $w \in \mathbb{R}^k$, $W_p \in \mathbb{R}^{k\times k}$ and $W_x \in \mathbb{R}^{k\times k}$ is a matrix with random initialization. $C_n$ is the LSTM's last output, max pooling or mean pooling, and $\alpha$ is the attention weight. $Rtt^* \in \mathbb{R}^k$ is the attention vector.

### 2.3 Text Similarity

There are many ways to calculate the similarity of text vectors, such as cosine distance, dot product and so on. In our system, we use a Bilinear way to calculate the similarity of attention vector and warrant vector. The formula is:

$$h = Rtt^* \times W_m \times W \tag{14}$$

where $Rtt^*$ is the attention vector, $W_m \in \mathbb{R}^{k\times k}$ is the randomly initialized weight matrix, and W is the warrant sentence vector that using LSTM's last output, max pooling or mean pooling.

### 2.4 Ensemble

Since neural networks have a large number of random parameters, we try to use different random initialization or change the network layer dimensions to adjust the network structure. In order to make the prediction more accurate, we run the program many times and use the voting method to obtain the final result.

### 2.5 Loss Function and Evaluation

We treat this task as a classification problem, and use log-loss as our loss function. The format is:

$$log - loss = \sum_{i=1}^{n} y_i log(h_i) + (1 - y_i) log(1 - h_i) \tag{15}$$

where $y_i$ is the label of i'th instance, and $h_i$ is the probability calculated by the system.

We also treat it as a sort problem, and choosing the top 1 of sorting results as the answer. The loss function format is:

$$loss = \sum_{i=1}^{n} max(0, 1 - sim(r, wa) + sim(r, w)) \tag{16}$$

where $sim(r, wa)$ is the true similarity of the premise and the warrant, and $sim(r, w)$ is the false similarity of the premise and the warrant.

Systems will be scored using accuracy. The format is:

$$accuracy = \frac{correct \quad predictions}{all \quad instances} \tag{17}$$

## 3 Experiments and Results

Table 1 shows the parameter setting in our system. Because we use Tensorflow to build our system, the sentence needs to be set to a fixed length. The sentences with length greater than 30 words are

| lstm_input_unit | lstm_output_unit | lstm_input_dropout | lstm_output_dropout | epoch |
|---|---|---|---|---|
| 300 | 200 | 0.6 | 0.6 | 40 |

Table 1: parameter setting in ITNLP_ARC system.

| | Train acc | Dev acc | Test acc |
|---|---|---|---|
| lstm(last-output)+seq-attention | 0.7450 | 0.6875 | 0.5315 |
| lstm(max-pooling)+seq-attention | 0.7519 | 0.6718 | 0.5382 |
| lstm(mean-pooling)+seq-attention | 0.8154 | 0.6906 | 0.5427 |
| lstm(last-output)+attention | 0.7737 | 0.6878 | 0.5257 |
| lstm(max-pooling)+attention | 0.7842 | 0.6827 | 0.5372 |
| lstm(mean-pooling)+attention | 0.7860 | 0.6932 | 0.5375 |

Table 2: The accuracy with log-loss on Semeval 2018 data sets.

| | Train acc | Dev acc | Test acc |
|---|---|---|---|
| lstm(last-output)+seq-attention | 0.7926 | 0.6841 | 0.5292 |
| lstm(max-pooling)+seq-attention | 0.7838 | 0.6812 | 0.5395 |
| lstm(mean-pooling)+seq-attention | 0.8360 | 0.6927 | 0.5495 |
| lstm(last-output)+attention | 0.7871 | 0.6750 | 0.5270 |
| lstm(max-pooling)+attention | 0.7929 | 0.6812 | 0.5225 |
| lstm(mean-pooling)+attention | 0.8105 | 0.6906 | 0.5427 |

Table 3: The accuracy with sort loss function om Semeval 2018 data sets.

| | Train acc | Dev acc | Test acc |
|---|---|---|---|
| Ensemble | 0.8319 | 0.7246 | 0.5521 |

Table 4: The accuracy of ensembling all neural network model.

truncated from the back, with length less than 30 words are added 0 in the behind.

In our system, we build the argument reasoning comprehension task with neural networks. We try to use the LSTM's last output, max pooling or mean pooling to represent the sentence vector, and use two kinds of attention to merge the reason and the claim. Because of neural networks contains a lot number of randomly initialized parameters, we run our system ten times and average the accuracy. Table 2 shows the accuracy with log-loss function. Table 3 shows the accuracy with sort loss function. From Table 2 and Table 3, we can get conclusion that mean pooling performed better than last output and max pooling. Table 4 shows the accuracy ensemble all neural network model, and this is our system's final result.

## 4 Conclusion and Future Works

We propose a neural network model to solve reasoning in NLP. We use attention model and bilinear to calculate the similarity between the premise and the warrant. Our system's final result achieved 0.5521. From the experiment, we can see the train accuracy and the development accuracy is much higher than test accuracy. This may be due to over fitting. Maybe decreasing learning rate, and using batch normalization can reduce over fitting. We will try it in the future work.

## Acknowledgment

This work is sponsored by the National High Technology Research and Development Program of China (2015AA015405) and National Natural Science Foundation of China (61572151 and 61602131).

## References

Yoshua Bengio, Patrice Y. Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks* 5(2):157–166. https://doi.org/10.1109/72.279181.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. pages 632–642. http://aclweb.org/anthology/D/D15/D15-1075.pdf.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. pages 1657–1668. https://doi.org/10.18653/v1/P17-1152.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 1724–1734. http://aclweb.org/anthology/D/D14/D14-1179.pdf.

Xinjian Gao, Tingting Mu, John Yannis Goulermas, and Meng Wang. 2018. Attention driven multimodal similarity learning. *Inf. Sci.* 432:530–542. https://doi.org/10.1016/j.ins.2017.08.026.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2017. The argument reasoning comprehension task. *CoRR* abs/1708.01425. http://arxiv.org/abs/1708.01425.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*. pages 1045–1048. http://www.isca-speech.org/archive/interspeech_2010/i10_1045.html.

Tomas Mikolov, Stefan Kombrink, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic*. pages 5528–5531. https://doi.org/10.1109/ICASSP.2011.5947611.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014a. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 1532–1543. http://aclweb.org/anthology/D/D14/D14-1162.pdf.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014b. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 1532–1543. http://aclweb.org/anthology/D/D14/D14-1162.pdf.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *CoRR* abs/1509.06664. http://arxiv.org/abs/1509.06664.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Sen Wang, and Chengqi Zhang. 2018. Reinforced self-attention network: a hybrid of hard and soft attention for sequence modeling. *CoRR* abs/1801.10296. http://arxiv.org/abs/1801.10296.

Hai-Tao Zheng, Wei Wang, Wang Chen, and Arun Kumar Sangaiah. 2018. Automatic generation of news comments based on gated attention neural networks. *IEEE Access* 6:702–710. https://doi.org/10.1109/ACCESS.2017.2774839.